

FIG. 1

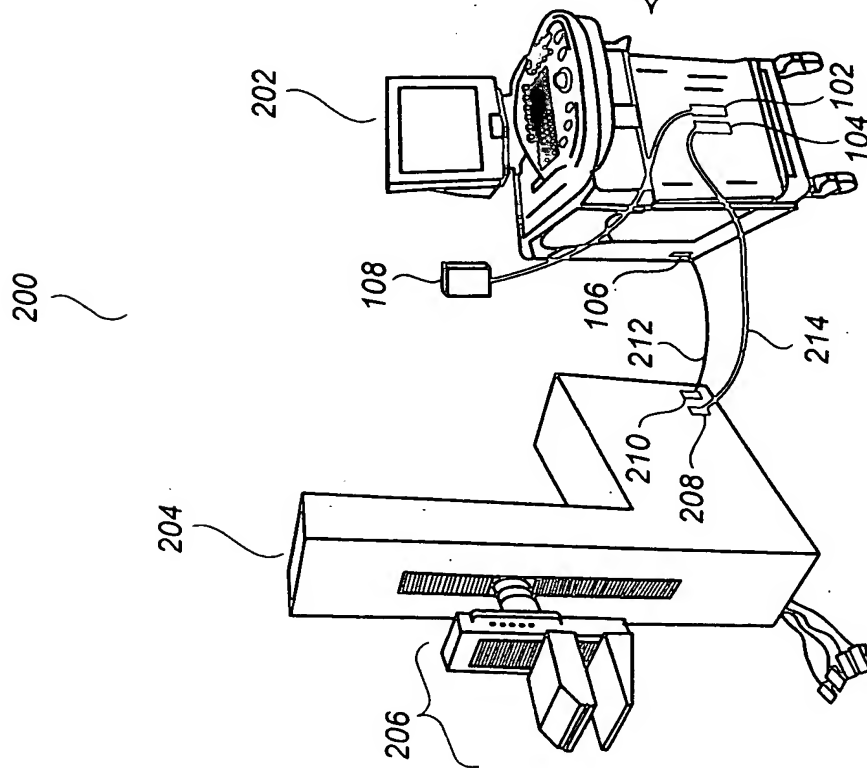
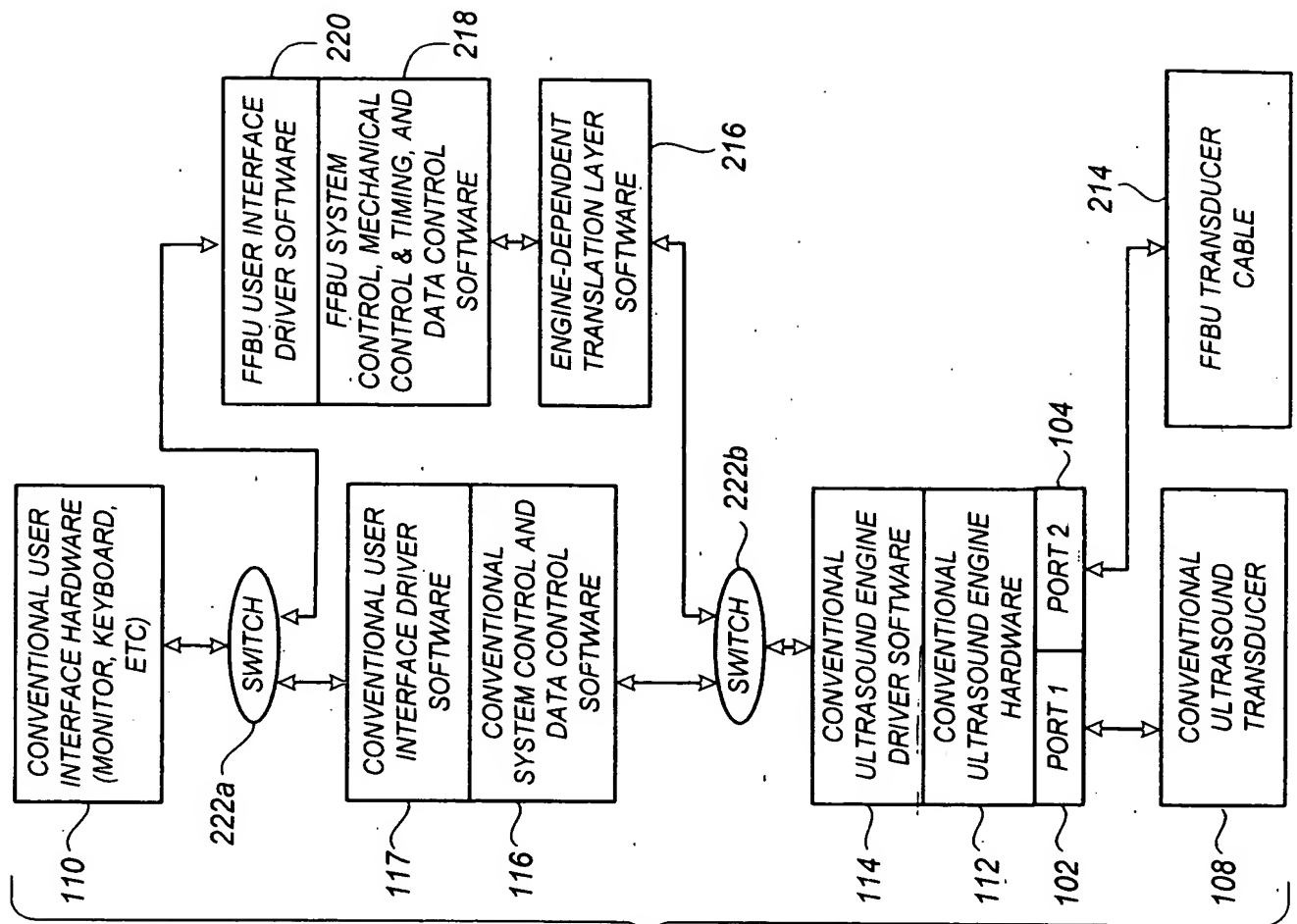


FIG. 2

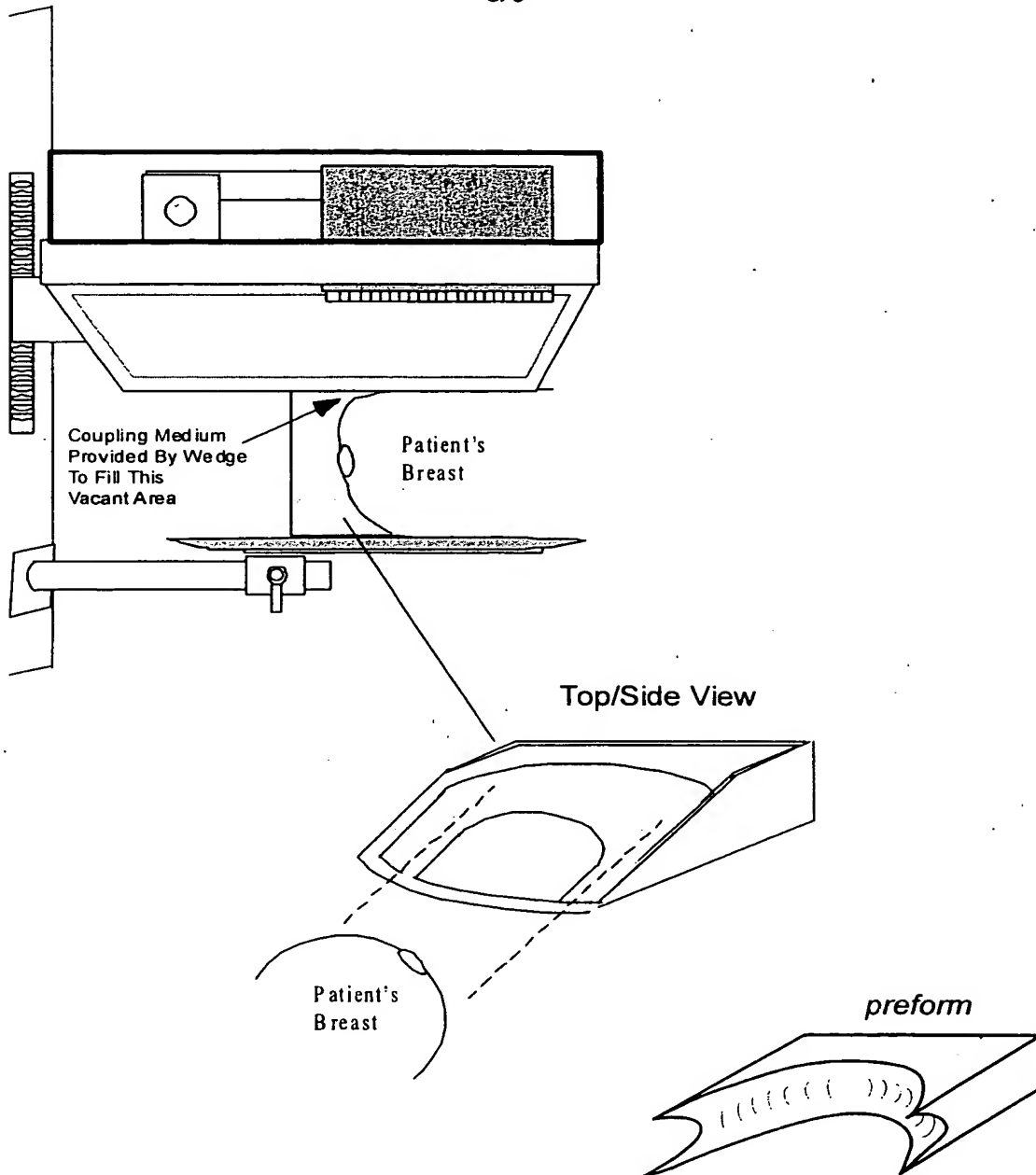


FIG. 3

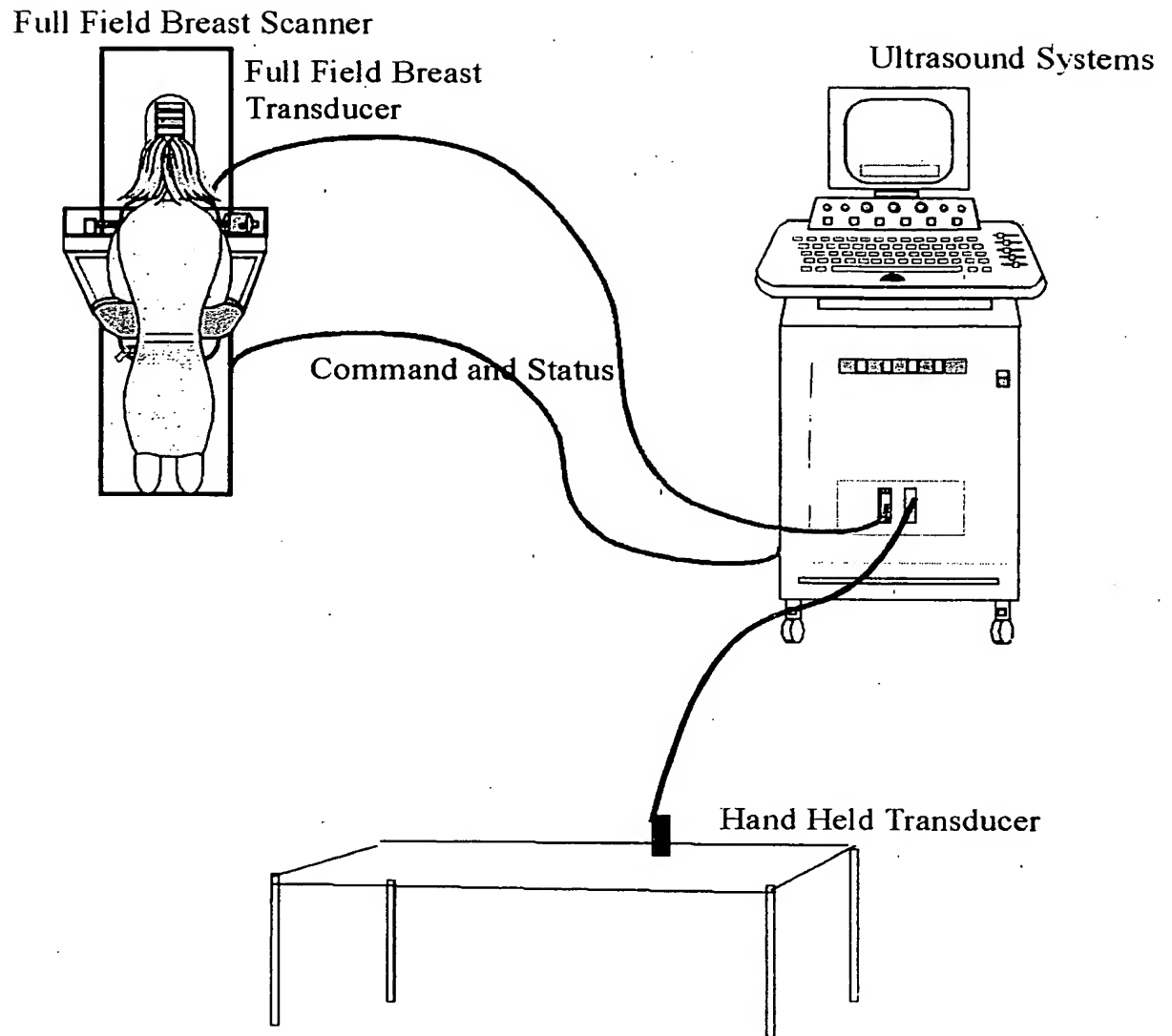


FIG. 4

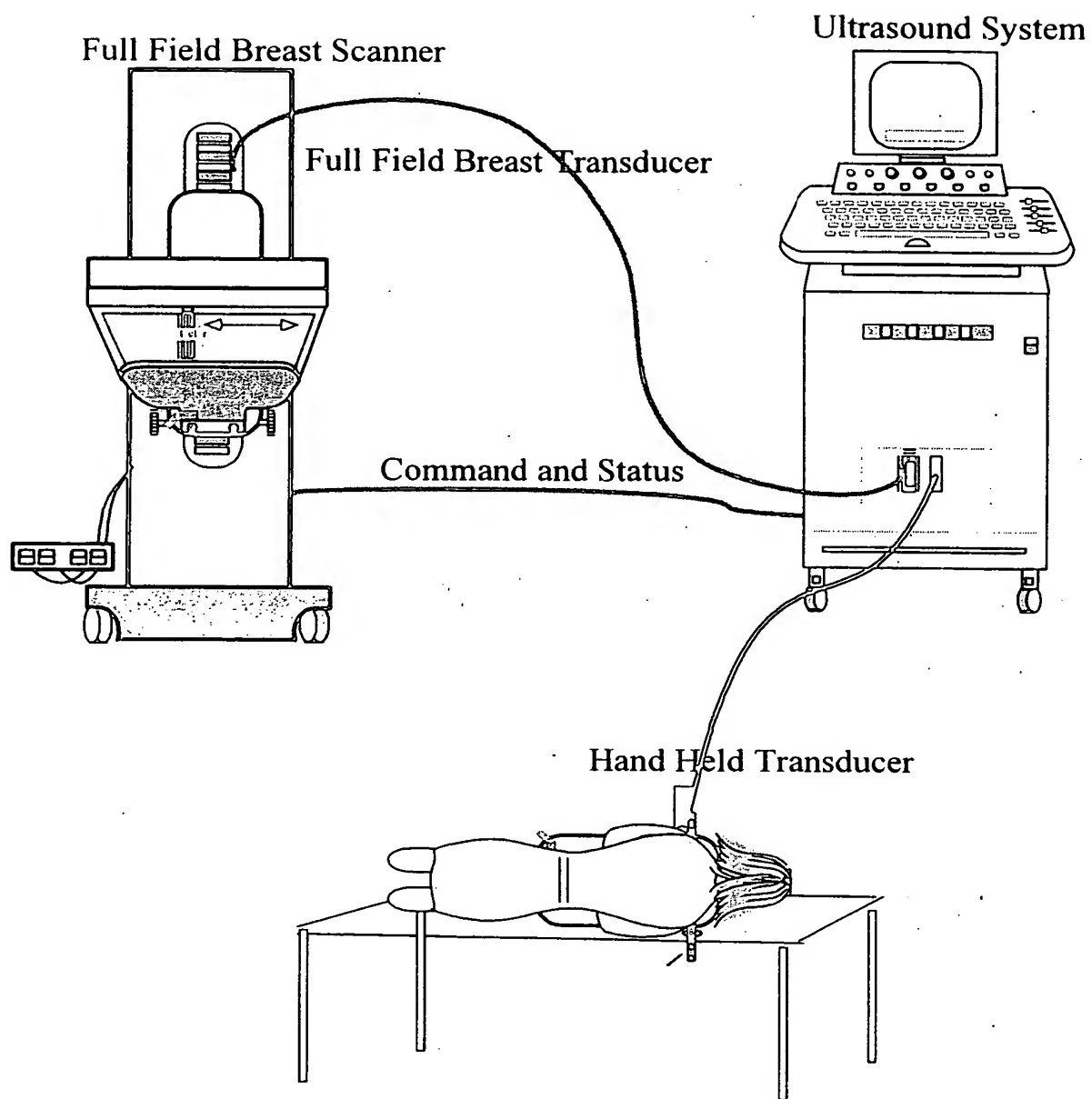
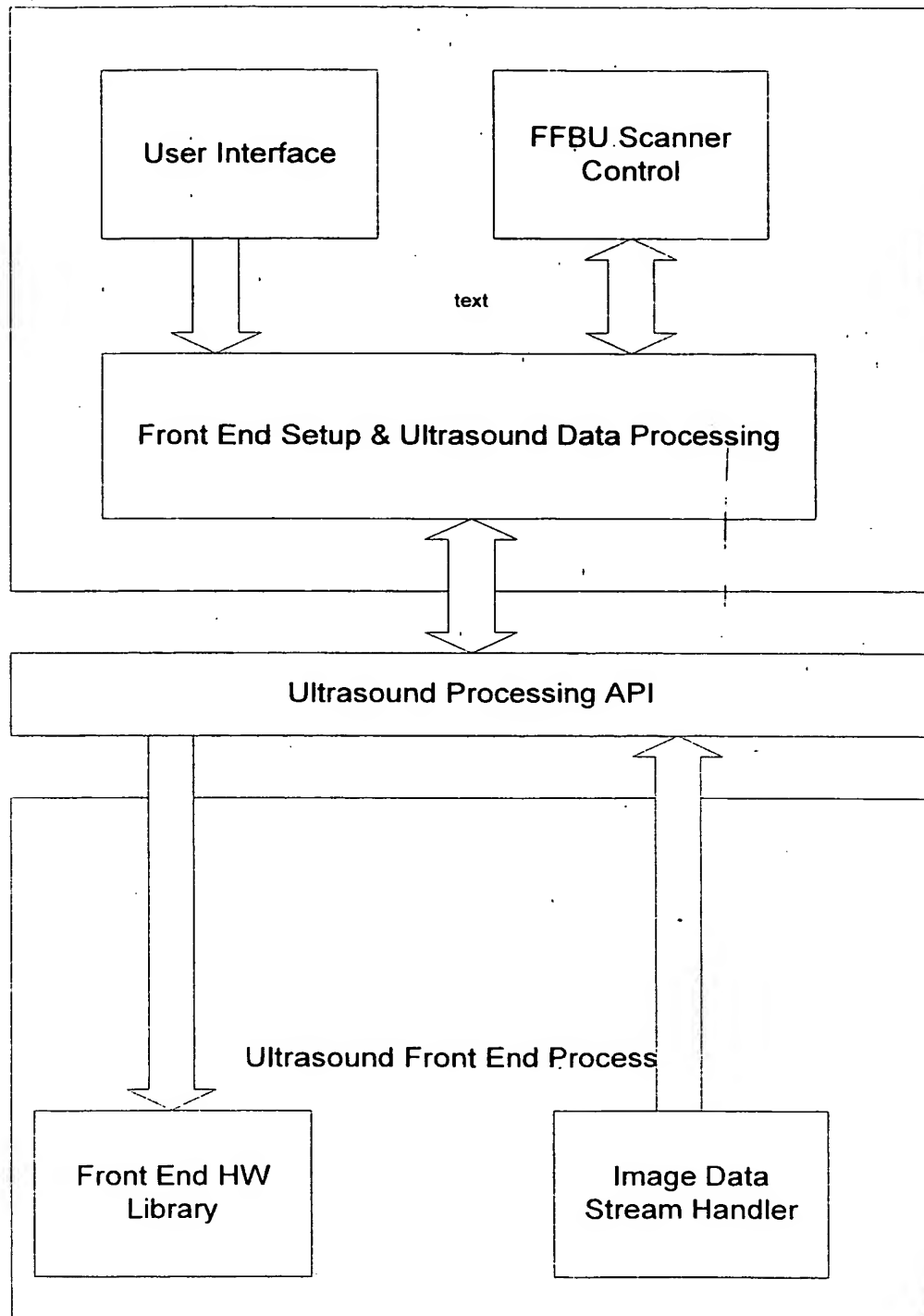


FIG. 5

**FIG. 6**

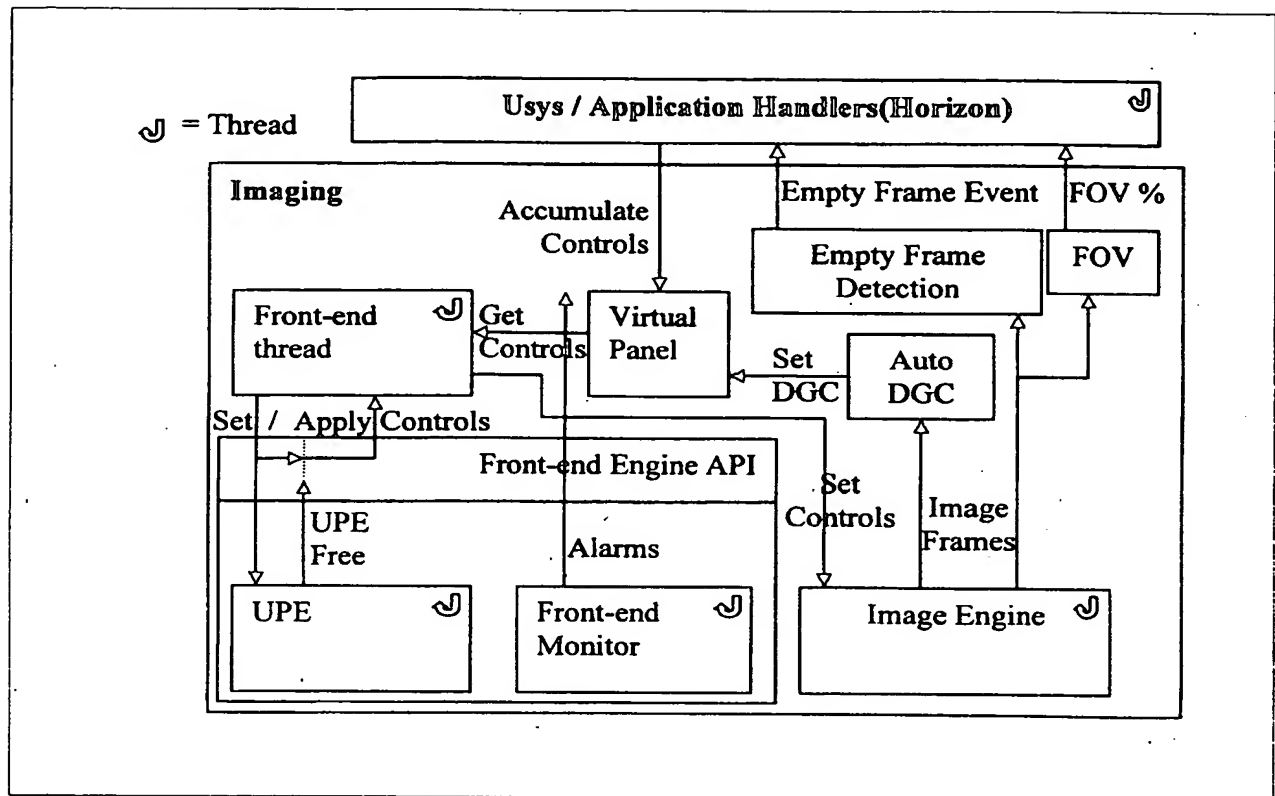


FIG. 7

Type Definitions

```

enum ElmagingModes { kModeB, kModeBColorFlow, kBModeColorPower,
    ... };
enum ElmagingControls { kDGC1, ..., kDGCn, kGainB, kDepth, kFOV,
    kFOVMode, ... };
enum EConvertModes { kNearestNeighbor, kBilinearInterpolation, ... };
struct SProbeParameters { std::string sProbeName, int iNumElements, double
    dElementPitchMM, ... };
struct SAlarmStatus { BOOL PowerStatus, int iCageTemperature, ... };

```

Imaging Control

```

Initialize();
    Initializes the imaging system, including loading the probe
    tables and establishing communication to the embedded system.
Terminate();
    Shuts down communication to the embedded system and powers
    it off.
vector< std::string> ReadProbeConnectorStatus();
    Retrieves a vector of length equal to the number of probe
    connectors. Each entry contains the name of the probe installed
    in that connector or "" if no probe is installed.
SelectProbeConnector( int iConnector );
    Enables the probe in the specified connector to acquire images
    and enables the probe table for that probe.
SProbeParameters GetProbeParameters( std::string sProbeName );
    Returns the probe parameters for the probe with the specified
    ProbeId.
int GetActiveProbeConnector( );
    Returns the connector index of the currently enabled probe.
vector< std::string> GetApplicationList( std::string sProbeName );
    Returns the list of the applications supported for the specified
    probe ID.

```

FIG. 8-1

Imaging Control (cont.)

```

vector< std::string> GetSubApplicationList( std::string sProbeName,
    std::string sApplicationName );
    Returns the list of the sub applications supported for the
    specified probe ID and application.
SelectProbeApplication( std::string sApplicationName, std::string
    sSubApplicationName );
    Selects the probe program associated with the specified
    program name for the currently selected probe.
SetImagingMode( EImagingModes );
    Set the imaging mode to the specified mode.
EImagingModes GetImagingMode( );
    Returns the currently active imaging mode.
SetImagingControl( EImagingControls elmagingControl, double dValue );
    Sets the specified user imaging control to the specified value.
ApplyImagingControls( );
    Applies the current set of imaging controls to the imaging HW.
    Blocks until the parameters are completely applied to the
    imaging HW.
double GetImagingControl( EImagingControls elmagingControl );
    Returns the current setting of the specified imaging control.
const SDisplayParameters GetDisplayParameters( );
    Read the current set of derived display parameters.
Freeze( BOOL bFreeze );
    Immediately freezes or unfreezes imaging, retaining the current
    imaging mode. Unfreeze will automatically apply any controls
    that are still pending.
BOOL GetFreezeStatus( );
    Returns the current freeze state.
SAlarmStatus GetAlarmStatus( );
    Returns the current front end alarm status.
Attach( Aspect aNotificationAspect );
    Attaches for notification on the specified aspect. Aspects
    include:
    Probe Inserted Or Removed
    Front End Alarm

```

Diagnostics / Engineering Tools

```

LoadProbeFile( );
    Reloads and applies the probe table for the active probe using the
    currently selected probe program.

```

FIG. 8-2